

1. Wstęp

Program obsługuje rezerwacje hotelowe, pozwala na ich dodawanie oraz usuwanie, przypisywanie pokoju do danej rezerwacji, zarządzanie klientami (dodawanie, usuwanie, przypisywanie rezerwacji), sprawdzanie dostępności pokoju w danych terminach. Służyć ma do zarządzania rezerwacjami, a więc przeznaczony jest dla pracowników hotelowych, m.in. recepcjonistów.

2. Architektura

System rezerwacji pokoi składa się z kilku głównych elementów, które razem umożliwiają wygodne zarządzanie pokojami, klientami i rezerwacjami w hotelu lub pensjonacie. Każdy element ma swoją konkretną rolę:

“Klienci” - to dane o osobach, które korzystają z hotelu.

System przechowuje ich imię, nazwisko i telefon, aby później łatwo było sprawdzać, kto zarezerwował dany pokój.

“Pokoje” - to lista pokoi dostępnych w hotelu.

Każdy pokój ma:

- swój numer,
- liczbę miejsc do spania,
- informację, ile jest w nim łóżek pojedynczych i podwójnych.

“Rezerwacje” - to najważniejszy element systemu — zawiera informacje o:

- numerze pokoju,
- danych klienta (np. numer telefonu),
- dacie początku rezerwacji,
- dacie zakończenia rezerwacji.

“Widoki” - System posiada cztery główne ekrany, które pozwalają użytkownikowi (np. recepcjoniście) sprawnie obsługiwać hotel:

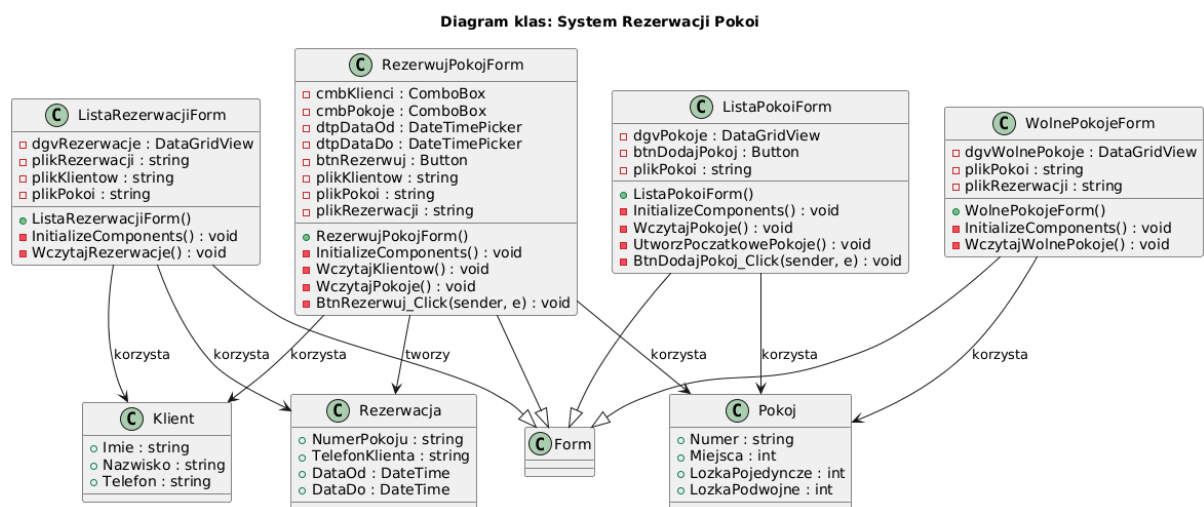
- Lista Rezerwacji — wyświetla wszystkie rezerwacje razem z informacjami o kliencie i terminie pobytu.
- Rezerwuj Pokój — pozwala wybrać pokój i klienta, a następnie zapisać nową rezerwację.
- Wolne Pokoje — prezentuje listę pokoi, które w danym momencie są dostępne.
- Lista Pokoi — wyświetla wszystkie pokoje dostępne w hotelu.

“Pliki danych” - System przechowuje dane w prostych plikach tekstowych:

- klienci.txt — dane klientów,
- pokoje.txt — dane pokoi,
- rezerwacje.txt — dane rezerwacji.

Te pliki pozwalają systemowi pamiętać i pokazywać wcześniej wprowadzone dane.

Schemat UML klas:



3. Klasy

“DodajKlienta” to formularz w aplikacji, który służy do dodawania nowego klienta. Pozwala użytkownikowi wpisać dane klienta i zapisać je do pliku tekstowego.

Odpowiada za:

- Wyświetlanie formularza z polami: Imię, Nazwisko, Telefon, Email.
- Umożliwianie zapisu danych klienta po kliknięciu przycisku „Zapisz”.
- Sprawdzenie, czy klient z takim telefonem lub e-mailem już istnieje, zanim doda go do pliku.
- Zapis dane do pliku tekstowego “klienci.txt”, oddzielając dane średnikami (;).

Zmienne w klasie:

- “TextBox txtImie, txtNazwisko, txtTelefon, txtEmail” – pola tekstowe, gdzie użytkownik wpisuje dane.
- “Button btnZapisz” – przycisk, który zapisuje dane klienta po kliknięciu.

- “string plikKlientow” = “klienci.txt” – ścieżka do pliku, gdzie zapisywane są dane klientów.

Metody:

“InitializeComponents()”

- Tworzy i ustawia wygląd elementów graficznych (etykiety, pola, przycisk).
- Ustawia tytuł okna, rozmiar, kolor tła itp.
- Podłącza przycisk do akcji: btnZapisz.Click += BtnZapisz_Click.

“BtnZapisz_Click(...)”

- Sprawdza, czy wszystkie pola są wypełnione.
 - Jeśli nie → pokazuje komunikat o błędzie.
 - Sprawdza, czy telefon lub email już istnieje w pliku.
 - Jeśli istnieje → pokazuje błąd i nie dodaje klienta.
 - Jeśli wszystko jest OK → dane zostaną dodane do pliku (ten fragment kodu zapewne będzie dalej, ale jeszcze go nie wysłałeś).
-

“**DodajPokoje**” to formularz w aplikacji, który służy do dodawania nowego pokoju hotelowego. Użytkownik wypełnia pola dotyczące pokoju i zapisuje te informacje do pliku tekstowego pokoje.txt.

Odpowiada za:

- Wyświetlenie formularza z polami do wypełnienia: numer pokoju, liczba miejsc, łóżka pojedyncze, łóżka podwójne.
- Sprawdzenie, czy wszystkie dane zostały wpisane.
- Zapisywanie danych do pliku pokoje.txt.

Zmienne w klasie:

- “txtNumer” – pole tekstowe do wpisania numeru pokoju.
- “txtMiejsca” – liczba wszystkich miejsc (np. 3 osoby).
- “txtLozkaPojedyncze” – ile jest łóżek pojedynczych.
- “txtLozkaPodwojne” – ile jest łóżek podwójnych.
- “btnZapisz” – przycisk, który po kliknięciu zapisuje dane.
- “plikPokoje” – nazwa pliku, do którego zapisywane są pokoje (pokoje.txt).

Metody:

“DodajPokojeForm()” – konstruktor:

- Tworzy nowe okno i ustawia jego wygląd przez wywołanie `InitializeComponents()`.

“InitializeComponents()” – inicjalizacja formularza:

- Tworzy wszystkie pola, etykiety i przycisk.
- Ustawia ich pozycje, rozmiary i wygląd.
- Dodaje je do formularza (czyli do okna aplikacji).
- Podłącza przycisk `btnZapisz` do metody, która zapisuje dane (`BtnZapisz_Click`).

“`BtnZapisz_Click(...)`” – co się dzieje po kliknięciu „Zapisz”:

- Sprawdza, czy użytkownik wypełnił wszystkie pola.
 - Jeśli czegoś brakuje → pokazuje komunikat o błędzie.
 - Jeśli wszystko jest OK: tworzy jeden tekstowy wiersz z danymi, oddzielonymi średnikami, dopisuje ten wiersz do pliku `pokoje.txt`, pokazuje komunikat „Pokój został dodany!”, zamyka formularz.
-

“**DodajRezerwacje**” to okno w aplikacji graficznej, które służy do tworzenia nowej rezerwacji pokoju przez klienta na konkretny termin.

Odpowiada za:

- Wyświetlenie formularza, w którym można wybrać:
 - klienta (z listy wczytanej z pliku `klienci.txt`),
 - pokój (z listy wczytanej z pliku `pokoje.txt`),
 - datę rozpoczęcia i zakończenia rezerwacji.
- Sprawdzenie, czy dane są poprawnie wypełnione.
- Zapisanie nowej rezerwacji do pliku `rezerwacje.txt`.

Zmienne w klasie:

- “plikKlientow, plikPokoi, plikRezerwacji” – ścieżki do plików z danymi.
- “klienci, pokoje” – listy przechowujące dane o klientach i pokojach.
- “cmbKlient, cmbPokoje” – listy rozwijane do wyboru klienta i pokoju.
- “dtpDataOd, dtpDataDo” – pola do wyboru dat (kiedy rezerwacja się zaczyna i kończy).
- “btnZapisz” – przycisk do zapisania rezerwacji.

Metody:

“DodajRezerwacjeForm()” – konstruktor:

- Uruchamia okno, ustawia wygląd i wczytuje dane klientów i pokoi.

“InitializeComponents()” – ustawianie wyglądu formularza:

- Tworzy wszystkie elementy graficzne (etykiety, listy rozwijane, daty, przycisk).
- Ustawia ich rozmieszczenie i styl.
- Dodaje je do formularza.
- Przycisk Zapisz jest podpięty do metody BtnZapisz_Click.

“WczytajDane()” – wczytuje dane z plików:

- Z klienci.txt – ładuje dane klientów do listy i do rozwijanej listy (ComboBox):
- Wyświetla ich jako: Imię Nazwisko (Telefon).
- Z pokoje.txt – ładuje numery pokoi do drugiej listy rozwijanej.

“BtnZapisz_Click()” – zapisanie nowej rezerwacji:

- Sprawdza, czy coś zostało wybrane:
- Jeśli nie wybrano klienta lub pokoju → pokazuje błąd.
 - Sprawdza daty:
- Data „od” nie może być późniejsza niż „do”.
 - Tworzy zapis rezerwacji
 - Zapisuje rezerwację do pliku rezerwacje.txt.
 - Pokazuje komunikat „Rezerwacja została zapisana” i zamyka formularz.
-

4. Cykl życia obiektów

- **Klienci**

Dodawanie: Klienci pojawiają się w systemie, gdy ktoś zgłasza chęć rezerwacji. Dane takiej osoby (imię, nazwisko, telefon) zostają zapisane w pliku `klienci.txt`.

Odczyt: System może później odczytać te dane, aby pokazać je podczas rezerwacji.

Usuwanie: Klienci mogą być usunięci ręcznie (poza systemem) poprzez edycję pliku, gdy stracą swoją ważność.

- **Pokoje**

Dodawanie: Nowe pokoje pojawiają się w pliku `pokoje.txt`, gdy hotel je uruchamia.

Odczyt: System sprawdza ten plik przy wyświetlaniu listy pokoi i gdy chcemy dokonać rezerwacji.

Usuwanie: Pokój może być usunięty poprzez ręczną edycję pliku (np. gdy pokój jest zamykany).

- **Rezerwacje**

Dodawanie: Powstają, gdy użytkownik (np. recepcjonista) wybierze:

- Klienta,
 - Pokój,
 - Termin od/do,
- i potwierdzi rezerwację. Dane trafiają do pliku `rezerwacje.txt`.

Odczyt: System używa danych rezerwacji, aby:

- Pokazać listę aktywnych rezerwacji,
- Sprawdzić dostępność pokoi (czy pokój jest zajęty w wybranym terminie).

Usuwanie: Rezerwacja przestaje być aktywna, gdy:

- Minie data jej zakończenia (jest „zamknięta”),
- Albo gdy użytkownik ręcznie usunie ją poprzez modyfikację pliku.

5. Przykładowy przepływ działania systemu

• **Uruchomienie systemu**

- Otwiera się główny ekran i wczytuje listę rezerwacji z pliku rezerwacje.txt.
- Klienci i pokoje są odczytywani z plików klienci.txt i pokoje.txt.

• **Dodanie nowej rezerwacji**

- Użytkownik wybiera Klienta i Pokój.
- Podaje daty rezerwacji.
- System sprawdza, czy pokój jest dostępny (analizując plik rezerwacje.txt).
- Jeśli pokój jest wolny, nowa rezerwacja jest zapisywana.

• **Wyświetlenie listy rezerwacji**

- System odczytuje plik rezerwacje.txt i łączy dane Klienta (z pliku klienci.txt) z numerami pokoi.
- Wynik prezentowany jest użytkownikowi w przejrzystej tabeli.

• **Wyświetlenie wolnych pokoi**

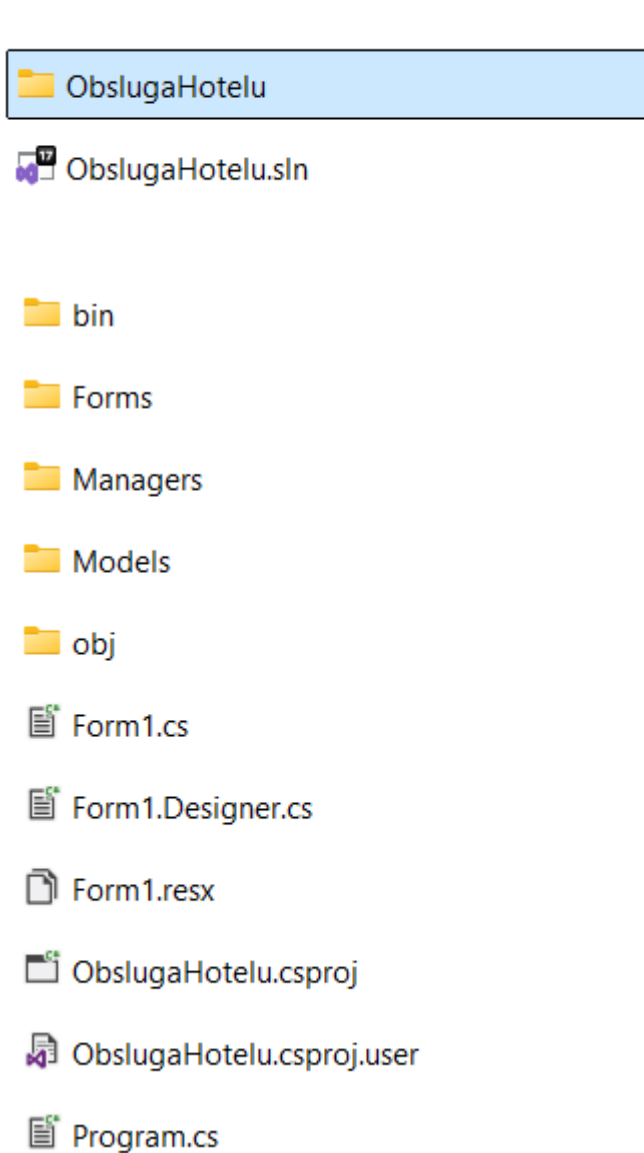
- System sprawdza plik rezerwacje.txt i filtruje listę pokoi (pokoje.txt) tak, aby pokazać te, które:
 - Nie mają aktywnej rezerwacji.
 - Są dostępne do wynajęcia.

- **Podsumowanie**

System działa jak sprawny notatnik:

- Klienci i pokoje to „zasoby” (zapisywane i odczytywane).
- Rezerwacje to „zamówienia” – mogą być dodane, sprawdzone i zakończone.
- Przepływ danych zapewnia, że zawsze mamy aktualny obraz tego, które pokoje i kiedy są dostępne.

6. Struktura plików



7. Autorzy

Michał Krawczyk

Michał Szymiczek